In the Claims

Please amend the claims as follows:

Claims 1-14 (canceled)

15. (Currently Amended) A computer implemented replacement selection method comprising:

creating a ~~binary~~ tree with a root node<u>, and</u> ~~and~~ multiple <u>non-root</u> ~~external~~ nodes, <u>and</u> <u>leaf nodes</u> ~~wherein a quantity of nodes in said tree is equal to a number of input data streams to be processed through said tree~~, each node having a first integer identifying one of at least three input data streams and a status identifier identifying a status of each of said input data streams, wherein a number of input streams is odd;

processing a data item from each of said at least three input streams by placing one data item from each of said three input stream<u>s</u> in each of said <u>leaf</u> ~~external~~ nodes of said tree, respectively;

promoting said data items of said <u>leaf</u> nodes through said ~~binary~~ tree <u>from a leaf node to said root node</u> ~~in a hierarchical manner~~, including comparing a first status identifier of a first node with a second status identifier of a second node, ~~and if said status identifiers are unequal~~ promoting one of said data items of one said nodes to a <u>node in a</u> next ~~hierarchical node~~ <u>level</u> in said tree <u>responsive to a status identifier value</u>, and switching a next comparison to a third status identifier of a third node with one of said other nodes in said tree;

~~compiling~~ <u>merging data items from said input streams into</u> a single output stream from said promoted data items, said output stream being a stream separate from said input streams, by copying compared data items promoted to said root node of said ~~binary~~ tree ~~and storing said single output stream on a computer readable medium~~, wherein the step of <u>merging</u> ~~compiling~~ said <u>data items into a</u> single output stream includes resolving comparison of duplicate identifiers ~~through an identification process~~ comprising:

comparing <u>said</u> ~~a~~ status identifier of two of said nodes;

<u>omitting a second node identifier comparison</u> ~~remembering at each node~~

2

when a first data item in a first f̶i̶s̶t̶ of said nodes remained in said node and was a duplicate of a second data item in a second of said nodes, wherein said second data item was promoted to a next hierarchical node in said tree; and

promoting said data item from said first node to a next hierarchical node in said tree absent a comparison of said data items in said nodes when a next time a said first node is designated for a key comparison; and

said single output stream comprised of said merged data items.

16. (Original) The method of claim 15, wherein the step of promoting said data item from said first node to a next hierarchical node in said tree absent a comparison of said data items includes skipping a comparison of said remembered first data item with data items in other nodes of said tree.

17. (Original) The method of claim 15, wherein said status identifier has a value corresponding to empty, duplicate, merging, and done.

18. (Currently Amended) The method of claim 17, wherein said status identifier is an integer variable, and said status identifier value corresponding to empty is a value of zero, said status identifier value corresponding to duplicate is a value of one, said status identifier value corresponding to merging is a value of two, and a status identifier value corresponding to d̶i̶n̶e̶ done is a value of three.

19. (Original) The method of claim 15, wherein the replacement selection method is a loser-oriented selection tree,

20. (Original) The method of claim 15, wherein the step of promoting said data item from said first node to a next hierarchical node in said tree absent a comparison of said data items includes avoiding exhausting promoting duplicate data items from a single input data stream.

3

21. (Currently Amended)   A system for processing computer-readable data comprising:

a ~~binary~~ tree with a root node, ~~and~~ multiple <u>non-root</u> ~~external~~ nodes, <u>and leaf nodes</u> ~~wherein a quantity of nodes in said tree is equal to a number of input data streams to be processed through said tree~~, each node having a first integer to identify one of at least three input data streams and a status identifier to identify a status of each of said input data streams;

a data item to be processed from each of said at least three input streams by placement of one data item from each of said three input stream in each of said <u>leaf</u> ~~external~~ nodes of said tree, respectively;

instructions to promote said data items of said <u>leaf</u> nodes through said ~~binary~~ tree <u>from a leaf node to said root node</u> ~~in a hierarchical manner~~, including comparison of a first status identifier of a first node with a second status identifier of a second node, ~~and if said status identifiers are unequal~~ to promote one of said data items of one said nodes to a ~~next hierarchical~~ node in <u>another level of</u> said tree <u>responsive to a status identifier value</u>, and to switch a next comparison to a third status identifier of a third node with one of said other nodes in said tree;

a <u>merge of said</u> ~~single output stream compiled from said promoted~~ data items~~, said output stream being a stream separate~~ from said input streams <u>into a single output stream from said promoted data items, said output stream being separate from said input streams</u> ~~by copy of compared data items promoted to said root node of said binary tree~~; and

~~a computer readable medium to store said compiled single output stream,~~ wherein ~~compilation~~ <u>merge</u> of said <u>input streams into said</u> single output stream includes~~:~~ instructions to resolve comparison of duplicate identifiers comprising:

comparison of a status identifier of two of said nodes;

<u>omission a second node identifier comparison</u> ~~remembering at each node~~ when a first data item in a first of said nodes remained in said node and was a duplicate of a second data item in a second of said nodes, wherein said second data item was promoted to a next hierarchical node in said tree; <u>and</u>

promotion of said data item from said first node to a next hierarchical node

4

in said tree absent a comparison of said data items in said nodes when a next time

a said first node is designated for a key comparison; and

said single output stream comprised of said merged data items.

22. (Currently Amended) The system of claim 21, wherein the instructions to promote said data item from said first node to a next hierarchical node in said tree absent a comparison of said data items includes skip of a comparison of said ~~remembered~~ first data item with data items in other nodes of said tree.

23. (Original) The system of claim 21, wherein said status identifier has a value corresponding to empty, duplicate, merging, and done.

24. (Previously Presented)   The system of claim 23, wherein said status identifier is an integer variable, and said status identifier value corresponding to empty is a value of zero, said status identifier value corresponding to duplicate is a value of one, said status identifier value corresponding to merging is a value of two, and a status identifier value corresponding to done is a value of three.

25. (Original) The system of claim 21, wherein the promotion instruction is a loser-oriented selection tree.

26. (Original) The system of claim 21, wherein the instructions to promote said data item from said first node to a next hierarchical node in said tree absent a comparison of said data items includes instructions to avoid exhausting promotion of duplicate data items from a single input data stream.

27. (Currently Amended)   An article comprising:

a computer-readable data storage medium;

instructions in the medium for implementing a replacement selection method

5

comprising:

instructions for creating a ~~binary~~ tree with a root node, ~~and~~ multiple ~~external~~ <u>non-root</u> nodes, <u>and leaf nodes,</u> ~~wherein a quantity of nodes in said tree is equal to a number of input data streams to be processed through said tree,~~ each node having a first integer identifying one of at least three input data streams and a status identifier identifying a status of each of said input data streams;

instructions for processing a data item from each of said at least three input streams by placing one data item from each of said three input stream<u>s</u> in each of said ~~external~~ <u>leaf</u> nodes of said tree, respectively;

instructions for promoting said data items of said <u>leaf</u> nodes through said ~~binary~~ tree <u>from a leaf node to said root node</u> ~~in a hierarchical manner~~, including comparing a first status identifier of a first node with a second status identifier of a second node, ~~and if said status identifiers are unequal~~ promoting one of said data items of one said nodes to a ~~next hierarchical~~ node in <u>a next level in</u> said tree <u>in response to a status identifier value,</u> and switching a next comparison to a third status identifier of a third node with one of said other nodes in said tree; ~~and~~

instructions for <u>merging data items from said input streams into</u> ~~compiling~~ a single output stream from said promoted data items, said output stream being a stream separate from said input streams, by copying compared data items promoted to said root node of said ~~binary~~ tree, ~~and storing said compiled single output stream on a computer readable medium;~~ wherein the instructions for ~~compiling~~ <u>merging</u> said <u>data items into a</u> single output stream includes instructions for resolving comparison of duplicate identifiers ~~through an identification process~~ comprising:

comparing <u>said</u> ~~a~~ status identifier of two of said nodes;

<u>omitting a second node identifier comparison</u> ~~remembering~~ at each node when a first data item in a first of said nodes remained in said node and was a duplicate of a second data item in a second of said nodes, wherein said second data item was promoted to a next hierarchical node in said tree; <u>and</u>

promoting said data item from said first node to a next hierarchical node in

6

said tree absent a comparison of said data items in said nodes when a next time a

said first node is designated for a key comparison; and

said single output stream comprised of said merged data items.

28. (Currently Amended) The article of claim 27, wherein the instructions for promoting said

data item from said first node to a next hierarchical node in said tree absent a comparison

of said data items includes skipping a comparison of said ~~remembered~~ first data item with

data items in other nodes of said tree.

29. (Original) The article of claim 27, wherein said status identifier has a value corresponding to

empty, duplicate, merging, and done.

30. (Previously Presented) The article of claim 29, wherein said status identifier is an integer

variable, and said status identifier value corresponding to empty is a value of zero, said

status identifier value corresponding to duplicate is a value of one, said status identifier

value corresponding to merging is a value of two, and a status identifier value

corresponding to done is a value of three.

31. (Original) The article of claim 27, wherein the replacement selection method is a loser-

oriented selection tree,

32. (Original) The article of claim 27, wherein the instructions for promoting said data item from

said first node to a next hierarchical node in said tree absent a comparison of said

data items includes avoiding exhausting promoting duplicate data items from a

single input data stream.

7